

---

# **bearcart Documentation**

***Release 0.1.2***

**Rob Story**

July 06, 2015



<b>1</b>	<b>Concept</b>	<b>3</b>
1.1	Getting Started . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>5</b>



[Rickshaw.js](#) is a great JavaScript library built on D3 by the folks at Shutterstock for plotting timeseries. [Pandas](#) is a great Python library built by a number of outstanding folks in the open source community for creating timeseries. Bear, meet Cart.



---

## Concept

---

Pandas Series and DataFrames with DatetimeIndex goes in. Rickshaw.js comes out.

Bearcart is a small library for creating Rickshaw visualizations with Pandas timeseries data structures. It has a simple API, a number of plot types, and some really nice legends and tooltips thanks to the folks at Shutterstock.

Bearcart uses Jinja2 templating to create the output, and the files are simple HTML/CSS/JS that can be manipulated after the fact for your application.

Contents:

### 1.1 Getting Started

Let's plot some stocks and make a line chart. Get data with Pandas, make visualization with Bearcart:

```
import bearcart
import pandas as pd

html_path = r'index.html'
data_path = r'data.json'
js_path = r'rickshaw.min.js'
css_path = r'rickshaw.min.css'

#All of the following import code comes from Wes McKinney's book, Python
#for Data Analysis

import pandas.io.data as web

all_data = {}

for ticker in ['AAPL', 'GOOG']:
    all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2010', '1/1/2013')

price = pd.DataFrame({tic: data['Adj Close']
                      for tic, data in all_data.iteritems()})

vis = bearcart.Chart(price)
vis.create_chart(html_path=html_path, data_path=data_path,
                js_path=js_path, css_path=css_path)
```

Go take a look at [this bl.ock](#) for the interactive example with the tooltip and legend data selection.

Lets try more companies, and an area plot:

```
all_data = {}
for ticker in ['AAPL', 'GOOG', 'XOM', 'MSFT', 'INTC', 'YHOO']:
    all_data[ticker] = web.get_data_yahoo(ticker, '1/1/2012', '1/1/2013')
price = pd.DataFrame({tic: data['Adj Close']
                      for tic, data in all_data.iteritems()})

vis = bearcart.Chart(price, plt_type='area')
```

Interactive version [here](#). Finally, let's make a scatterplot with some custom colors:

```
df = pd.concat([price['AAPL'], price['GOOG']], axis=1)[:100]

vis = bearcart.Chart(df, plt_type='scatterplot', colors={'AAPL': '#1d4e69',
                                                         'GOOG': '#3b98ca' })
```

Interactive example [here](#)

If you don't want some of the chart features, like the legend, hover, x-axis, etc, you can just pass those parameters as false when defining the chart:

```
vis = bearcart.Chart(df, hover=False, legend=False)
```

That's it- a small little library for making nice little interactive timeseries charts. Happy plotting!



---

## Indices and tables

---

- `genindex`
- `search`